

Certified Kinematics Solution of 2-DOF Planar Parallel Continuum Mechanisms

Oscar Altuzarra¹ and Jean Pierre Merlet²

¹ University of the Basque Country UPV/EHU, Bilbao, Spain,
`oscar.altuzarra@ehu.es`

² INRIA, Sophia-Antipolis, France,
`Jean-Pierre.Merlet@sophia.inria.fr`

Abstract. The abstract should summarize the contents of the paper using at least 70 and at most 150 words. ... here we have a FK and IK that are somewhat difficult to solve if we are looking for all solutions. But let's remember that we have also a certified newton scheme for real time purpose.

Keywords: interval analysis, elliptic integrals, ...

1 Introduction

2 Quasi-static Equilibrium Equations

Planar parallel continuum mechanisms under consideration are those closed-loop assemblies of two flexible rods joined together at an end-effector point \mathcal{P} , and actuated at their extreme connected to the fixed frame. External loads are applied at the extremes of the rods, either loads at the end-effector or torques at the actuated ends.

As a way of example, the mechanism in Fig. 1 will be analysed in more detail. It is composed by two rods ($r = 1, 2$) of length L_r articulated together at \mathcal{P} , and actuated by rotational actuators placed at points O_r . Being its attachment a clamped condition, they control input angles α_r . Output pose $\mathbf{p}_{\mathcal{P}}$ of \mathcal{P} is given by coordinates in a fixed frame. Two moving frames are defined with fixed origin at O_r , and oriented with the clamping condition by α_r . It is denoted by $2\underline{R}FR$, where \underline{R} stands for revolute actuators, F for the flexible rod, and R for the revolute joint at \mathcal{P} .

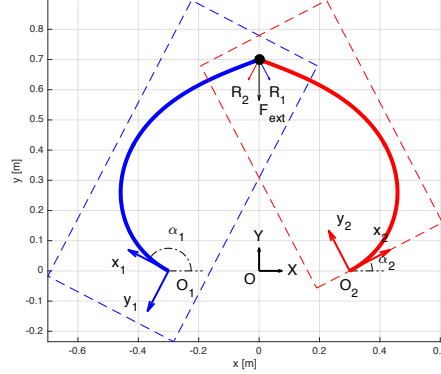


Fig. 1. A $2\underline{R}FR$ mechanism.

Any pose of the mechanism is a static equilibrium of the system. Hence, its punctual end-effector is in equilibrium under the external load $[F_X; F_Y]$ and the reaction forces due to the nonlinear deformation of the rods, given by modulus R^r and angle ψ^r with axis \mathbf{x}_r of the moving frame r . Namely, (**JPM comment:** I believe that the ψ, ϕ angles should be shown in the figure)

$$\sum_{r=1}^2 \mathbf{R}_r^F \cdot \begin{Bmatrix} R^r \cos \psi^r \\ R^r \sin \psi^r \end{Bmatrix} + \begin{Bmatrix} F_X \\ F_Y \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \quad (1)$$

where \mathbf{R}_r^F is the rotation matrix between the moving frame r and the fixed frame.

(**JPM comment:** Here we should indicate that this relation imposes to have the same R^r for both rods)

End point coordinates of each rod in their local frames are the result of a nonlinear analysis of its deformation. Transformations of the homogeneous coordinates of those end-points $\bar{\mathbf{p}}^r(L_r) = [x_r, y_r, 1]^T$ from local frames to the fixed frame require transformation matrices \mathbf{T}_r^F that depend on the inputs. As rods are joined together at \mathcal{P} , end-coordinates for both rods must be equal:

$$\mathbf{T}_1^F \bar{\mathbf{p}}^1(L_1) = \mathbf{T}_2^F \bar{\mathbf{p}}^2(L_2) \quad (2)$$

(JPM comment: I believe we should give the T matrix as function of α_r)

Whenever the output coordinates $\mathbf{p}_{\mathcal{P}}$ of \mathcal{P} are known, along with Eq. 2 the following also applies:

$$\mathbf{T}_r^F \begin{Bmatrix} x_r \\ y_r \\ 1 \end{Bmatrix} = \bar{\mathbf{p}}_{\mathcal{P}} \quad \text{for } r = 1, 2 \quad (3)$$

The nonlinear analysis of the deformation of such rods can be effectively solved making use of elliptic integrals. As a result, the local coordinates x_r and y_r of the deformed shape of the rod r for positions given by ϕ_i along the arc length are:

$$\begin{aligned} x_r(\phi_i) = & -\sqrt{\frac{EI}{R^r}} \cos \psi^r [2E(k^r, \phi_i) - 2E(k^r, \phi_1) - F(k^r, \phi_i) + F(k^r, \phi_1)] - \\ & -\sqrt{\frac{EI}{R^r}} 2k^r \sin \psi^r [\cos \phi_i - \cos \phi_1] \end{aligned} \quad (4)$$

$$\begin{aligned} y_r(\phi_i) = & -\sqrt{\frac{EI}{R^r}} \sin \psi^r [2E(k^r, \phi_i) - 2E(k^r, \phi_1) - F(k^r, \phi_i) + F(k^r, \phi_1)] + \\ & +\sqrt{\frac{EI}{R^r}} 2k^r \cos \psi^r [\cos \phi_i - \cos \phi_1] \end{aligned} \quad (5)$$

where E is the elastic modulus, I the moment of the cross-section, $F(k^r, \phi_i)$ and $E(k^r, \phi_i)$ are the incomplete elliptic integrals of the first and second kind, with k^r the modulus in the range $[-1; 1]$.

Angle ϕ varies continuously along the length of the rod between ϕ_1 and ϕ_2 , which are defined by the boundary conditions of the rod at extremes. At the first extreme, the horizontal clamped condition provides the corresponding value of ϕ , namely

$$\phi_1 = \arcsin \left(\frac{1}{k^r} \cos \left(\frac{\psi^r}{2} \right) \right) \quad (6)$$

whereas at the other extreme, a hinged condition, its null bending moment implies a null curvature and it defines a point of inflection of the curve representing the deformation, so ϕ at this position can be deduced to be:

$$\phi_2 = (2q - 1)\pi/2 \quad \text{for } q = 1, 2, \dots \quad (7)$$

where q identifies the so-called buckling modes. Each buckling mode has a corresponding number of points of inflection, i.e., mode 1 has one inflection point precisely at the hinged end, mode 2 has two, and so on.

(JPM comment: We must indicate somewhere that ϕ_1 is the ϕ^r that we will be looking for in the next sections while the ϕ_i will be ϕ_2 . This was unclear for me when I start working on the problem)

The value of the reaction force at the extreme of the rod to verify the force equilibrium of the rod is given by:

$$\sqrt{R^r} = \frac{\sqrt{EI}}{L_r} [F(k^r, \phi_2) - F(k^r, \phi_1)] \quad (8)$$

Each flexible rod r is ruled by Eqs. 4, 5 and 8, and uniquely defined by parameters k^r and ψ^r . Any quasi-static position problem of the deformed mechanism is solved by finding the set of parameters k^r and ψ^r for each rod subject to the corresponding boundary conditions. Such is the case of the mechanism in Fig. 1, whose solution corresponds to the set of k^r and ψ^r values shown in Fig. 2 for each of the rods.

On the one hand, parameter k^r is the modulus of the elliptic integrals, so limited to a range $[-1; 1]$, but restricted by the boundary condition in Eq. 6 so that real values are obtained. Hence, its range is limited to $[-1; -k_{min}^r]$ and $[k_{min}^r; 1]$ with:

$$k_{min}^r = \left| \cos \left(\frac{\psi^r}{2} \right) \right| \quad (9)$$

On the other hand, parameter ψ^r runs in a range $[0; 2\pi]$. Hence, any numerical algorithm in search of solutions for the problem has a closed parameter space to look at.

(JPM comment: For the sake of clarity I suggest to properly define the kinematic problem before going on)

3 Kinematic problems

3.1 Forward kinematic

In the forward kinematic problem the angles α_r are given and we have to determine the location(s) of the end-effector of the robot that are defined by the coordinates x_p, y_p of \mathcal{P} in the reference frame. The unknowns are the k^r, ψ^r, ϕ^r of each rod together for a total of 6 unknowns. If we assume that we know the buckling mode (i.e the value of q in equation (7)) the constraints are the 2 mechanical equilibrium conditions (1), the 2 equations of (3) that are used to indicate that each rod ends at the same point, the two equations (6) that relate the ϕ^r, k^r, ψ^r of each rod for a total of 6 and constraints. We have therefore a square system that will usually admit only a finite set of solutions. Remember also that we are looking only for solutions such that $k^r \in [-1, 1]$, $\phi^r \in [-\pi/2, \pi/2]$ and $\psi^r \in [0, 2\pi]$.

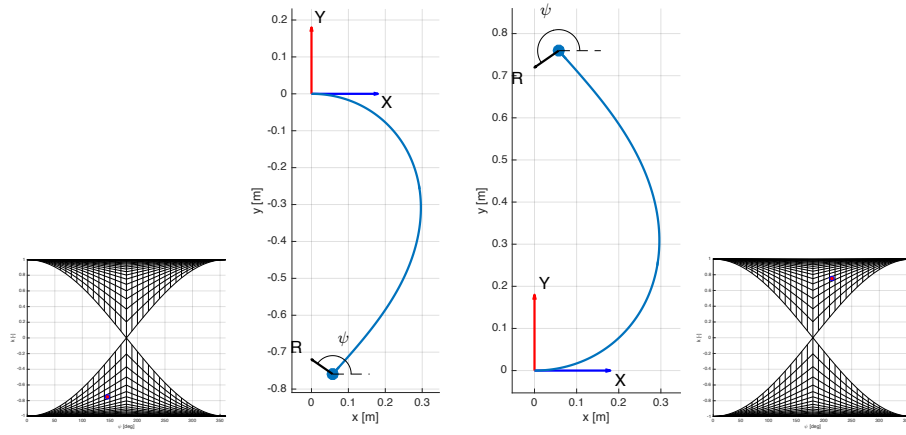


Fig. 2. Parameter values for rods in the $\underline{2RRR}$ mechanism.

3.2 Inverse kinematics

For this problem the coordinates x_p, y_p are known and we have to determine the possible values of the 2 α_r . Again if we assume that we know the buckling mode the unknowns are the 6 k^r, ϕ^r, ψ^r and the two α_r for a total of 8 unknowns. The constraints are the 2 mechanical equilibrium conditions (1) and the two equations (6) that relate the ϕ^r, k^r, ψ^r of each rod. Additionally (3) will provide 2 equations for each rod so that we have a total of 8 constraints so that the inverse kinematic is a square problem with a finite number of solutions.

4 Solving with Residual Analysis in Parameter Space

A first approach to the finding of all solutions of the kinematics problem above is a comprehensive search of the four-dimensional parameter space k^r - ψ^r , looking for those sets of parameters that fulfil the corresponding geometric and static equilibrium conditions. First, a redefinition of the parameter space is presented. Second, as some differences apply for the Forward Kinematics (FK) with respect to the Inverse Kinematics (IK), they will be stated differently.

As the range for k^r is dependent on ψ^r as shown in Eq.9, we have defined an alternative to the parameter k^r as:

$$k_{rel}^r = \text{Sign}(k^r) \cdot \frac{|k^r| - |k_{min}^r|}{1 - |k_{min}^r|} \quad (10)$$

The change in the parameter space can be observed in Fig. 3, and allows an easier grid definition with a finer mesh whereas involved functions have sharper changes.



Fig. 3. Plots of feasible values of k^r against ψ^r versus k_{rel}^r against ψ^r .

In the FK, input angles α_r are known, then rotation and transformation matrices \mathbf{R}_r^F and \mathbf{T}_r^F are defined. The static equilibrium condition in Eq. 1 and the assembly of end-tips of rods in Eq. 2 must be fulfilled. Those conditions define the deformed shape of the mechanism, i.e. k^r - ψ^r . The output pose \mathbf{p}_P is obtained afterwards from end-tip coordinates. The process to follow is stated in algorithm 1.

Algorithm 1: Full Forward Kinematics

Data: input angles α_r
Result: deformed shape of rods k^r - ψ^r and output pose $\mathbf{p}_{\mathcal{P}}$
 initialization;
 Find matrices \mathbf{R}_r^F and \mathbf{T}_r^F ;
for $k_{rel}^1 = -1$ **to** 1 **do**
 for $\psi^1 = 0$ **to** 2π **do**
 Find reaction force R^1 in Eq. 8;
 Solve R^2 and ψ^2 in Eq. 1;
 Find $\bar{\mathbf{p}}^1(L_1) = [x_1, y_1, 1]^T$ with Eqs. 4 and 5;
 Find end-tip coordinates in fixed frame: $\bar{\mathbf{p}}^F(L_1) = \mathbf{T}_1^F \bar{\mathbf{p}}^1(L_1)$;
 for $k_{rel}^2 = -1$ **to** 1 **do**
 Find R_k^2 in Eq. 8;
 Calculate $Residue_{R^2} = R_k^2 - R^2$;
 Find $\bar{\mathbf{p}}^2(L_2) = [x_2, y_2, 1]^T$ with Eqs. 4 and 5;
 Find end-tip coordinates: $\bar{\mathbf{p}}^F(L_2) = \mathbf{T}_2^F \bar{\mathbf{p}}^2(L_2)$;
 Calculate $Residue_{\mathbf{p}^F} = \bar{\mathbf{p}}^F(L_2) - \bar{\mathbf{p}}^F(L_1)$;
 end
 end
end
 Interpolate Null Residues and plot them as in Figs. 4 and 5 ;
 Intersect Null Residuals to get potential solutions k^1 - ψ^1 - k^2 - ψ^2 as in Fig. 6;
 Run Newton scheme to refine solutions;
 Provide output poses $\mathbf{p}_{\mathcal{P}} = \mathbf{p}^F$;
 Plot Solutions as in Fig. 6 ;

(JPM comment: If I understand well this algorithm is valid only for given buckling modes of the rod ? this must be indicated somewhere)

When running on the grid k_{rel}^1 - ψ^1 , we get the value for the end-tip force R^1 in Eq. 8. Upon substitution into the static equilibrium condition Eq. 1, we get the end-tip force of the second rod R^2 and ψ^2 . Hence, there is no need to run the analysis in the ψ^2 range, and the search is restricted now to the k_{rel}^2 . For the sampled values of the latter we can obtain the corresponding R^2 value with Eq. 8. Whenever the latter value for R^2 equals the one obtained first in the static equilibrium, we get sets of parameters k_{rel}^r - ψ^r that fulfil the static equilibrium condition. This can be found by interpolation of sampled results in the space k_{rel}^1 - ψ^1 - k_{rel}^2 and plotted as in Fig. 4.

For each of the same points considered above, we can get the rods' end-tip local coordinates in Eqs. 4 and 5, and evaluate the assembly condition in Eq. 2. The plots of the assembly fulfilment in X and Y are shown in Fig. 5.

Finally, the three null-residual conditions intersect at solutions to the FK problem, see Fig. 6-left. A refinement of each potential solution can be done

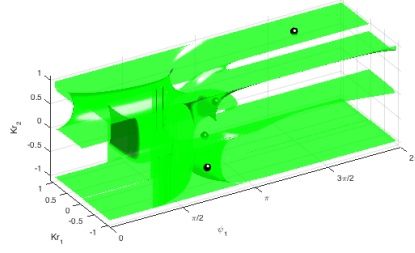


Fig. 4. Plot of fulfilment of Force balance in the parameter space k_{rel}^1 - ψ^1 - k_{rel}^2 .

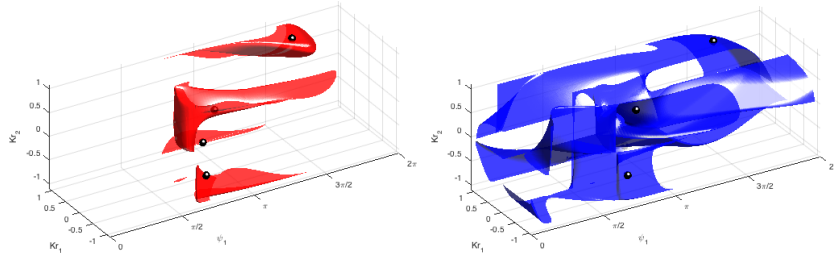


Fig. 5. Plot of fulfilment of assembly in the parameter space k_{rel}^1 - ψ^1 - k_{rel}^2 .

with a Newton scheme starting from those intersections, results can then be plotted as in Fig. 6-right.

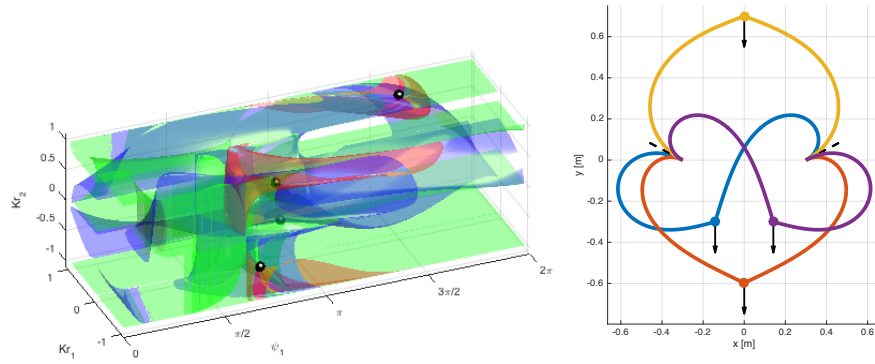


Fig. 6. Multiple FK Solutions in the parameter space k_{rel}^1 - ψ^1 - k_{rel}^2 .

For the IK, output pose of \mathcal{P} , i.e. $\mathbf{p}_{\mathcal{P}}$, is given, while input angles α_r are unknown. Rotation and transformation matrices \mathbf{R}_r^F and \mathbf{T}_r^F are functions of these unknowns. Therefore, the deformed shape of the mechanism, $k^r\text{-}\psi^r$, and the input angles α_r are the unknowns in a coupled system of equations defined by the static equilibrium condition in Eq. 1 and the given location of end-tips of rods in Eq. 3. Six conditions with six unknowns that would require a six-dimensional search in a restricted parameter space.

The problem is very much simplified if the following approach is followed. Let's consider the first rod isolated, with its base-end at O_1 and the tip-end at known output \mathcal{P} . For each α_1 value taken in the range $[0; 2\pi]$ we can search for the $k^1\text{-}\psi^1$ sets that accomplish the assembly condition with Eq. 3. Those sets produce a reaction force value R^1 in Eq. 8. With that value and the corresponding orientation in the fixed frame, static equilibrium condition in Eq. 1 can be stated to find the required force from rod 2 to accomplish equilibrium, i.e. $R^{2\text{ eq}}$ with orientation β_2 in the fixed frame, see Fig. 7-left.

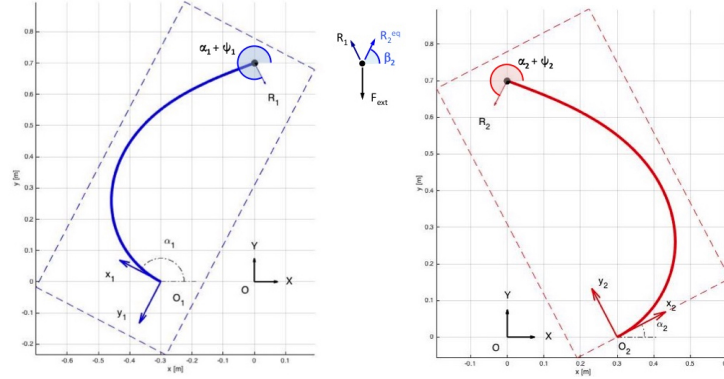


Fig. 7. IK Solutions search.

Now, we can run the same search in rod 2, varying α_2 in the range $[0; 2\pi]$ we can search for the $k^2\text{-}\psi^2$ sets that accomplish the assembly condition with Eq. 3. For those sets obtained, we can get the reaction force R^2 and its orientation in the fixed frame β_2 . If the values for the required $R^{2\text{ eq}}$ force and the obtained R^2 force are plotted against orientation β_2 , see Fig. 8, we get at intersections the IK solutions of the mechanism.

The process to follow can be stated as in algorithm 2.

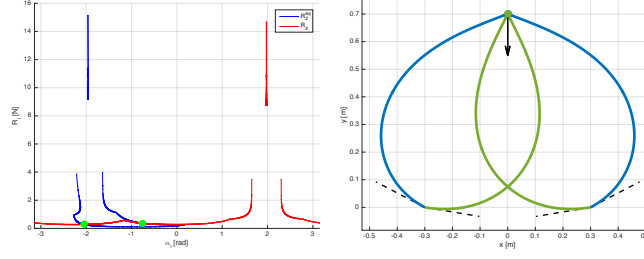


Fig. 8. IK finding scheme and solutions.

Algorithm 2: Full Inverse Kinematics

Data: output pose $\mathbf{p}_{\mathcal{P}}$
Result: deformed shape of rods k^r - ψ^r and input angles α_r

initialization;
for $\alpha_1=0$ **to** 2π **do**
 Find matrices \mathbf{R}_1^F and \mathbf{T}_1^F ;
 for $k_{rel}^1=-1$ **to** 1 **do**
 for $\psi^1=0$ **to** 2π **do**
 Find reaction force R^1 in Eq. 8;
 Find $\bar{\mathbf{p}}^1(L_1) = [x_1, y_1, 1]^T$ with Eqs. 4 and 5;
 Find end-tip coordinates in fixed frame: $\bar{\mathbf{p}}^F(L_1) = \mathbf{T}_1^F \bar{\mathbf{p}}^1(L_1)$;
 if $\bar{\mathbf{p}}^F(L_1) = \mathbf{p}_{\mathcal{P}}$ **then**
 Solve R^{2eq} and β_2 for equilibrium of \mathcal{P} ;
 end
 end
 end
end
for $\alpha_2=0$ **to** 2π **do**
 Find matrices \mathbf{R}_2^F and \mathbf{T}_2^F ;
 for $k_{rel}^2=-1$ **to** 1 **do**
 for $\psi^2=0$ **to** 2π **do**
 Find reaction force R^2 in Eq. 8;
 Find $\bar{\mathbf{p}}^2(L_2) = [x_2, y_2, 1]^T$ with Eqs. 4 and 5;
 Find end-tip coordinates in fixed frame: $\bar{\mathbf{p}}^F(L_2) = \mathbf{T}_2^F \bar{\mathbf{p}}^2(L_2)$;
 if $\bar{\mathbf{p}}^F(L_2) = \mathbf{p}_{\mathcal{P}}$ **then**
 Save R^2 and $\beta_2 = \alpha_2 + \psi^2 - \pi$;
 end
 end
 end
end
Plot R^{2eq} and R^2 versus β_2 as in Fig. 8;
Intersect curves to get potential solutions k^1 - ψ^1 - k^2 - ψ^2 ;
Run Newton scheme to refine solutions;
Provide input values α_1 and α_2 ;
Plot Solutions as in Fig. 8 ;

(**JPM comment:** I suggest to provide computation time here)

5 Solving with Interval Analysis

As seen in the previous section the residual approach provides solution in a low computation time. However it cannot guarantee to find **all** solutions. As seen previously the kinematic problems we are dealing with imply to look for solution only within a bounded region in the parameter space. Therefore we propose another approach that is convenient to deal with solving within a bounded region **and** to guarantee to find all possible solutions: interval analysis (IA) [?,?].

The primary element of interval analysis is based on the *interval evaluation* of a mathematical expression $F(x_1, x_2, \dots, x_n)$ in which the x_i have an interval value, meaning that all $x_i \in [\underline{x}_i, \overline{x}_i]$. This interval evaluation provides a range $[A, B]$ such that for any real-valued instance $\{x_1^r, x_2^r, \dots, x_n^r\}$ of the set of x_i that satisfy the range constraint we have

$$A \leq F(x_1^r, x_2^r, \dots, x_n^r) \leq B$$

In other words A, B are (possibly not optimal) minimum and maximum of F over the domain. There are multiple way to obtain an interval evaluation but the simplest one is the *natural evaluation* that consists in substituting all the mathematical operation of F by their interval equivalent. Numerous software packages provide such an interval evaluation, some of them even managing round-off errors so that the values of A, B are **guaranteed**. For example in the forward kinematic problem being given ranges for the k^r, ψ^r, ϕ^r of rod 1, we are able to compute an interval evaluation of x_r, y_r using equations (4, 5). Then using the fixed transformation of (3) we will get the ranges $[\underline{x}_p^1, \overline{x}_p^1], [\underline{y}_p^1, \overline{y}_p^1]$ for the x_p, y_p coordinates. In the same manner we may obtain the ranges $[\underline{x}_p^2, \overline{x}_p^2], [\underline{y}_p^2, \overline{y}_p^2]$ for rod 2. The most important point is that if the corresponding ranges for rod 1 and 2 do not intersect (e.g if $\underline{x}_p^1 > \overline{x}_p^2$, then the forward kinematic cannot have a solution within the provided ranges for the k, ϕ, ψ . A major drawback of interval arithmetic is however that A, B may be overestimated i.e A is much lower than the real minimum of F , while B is much larger than its maximum. A classical example of this *wrapping effect* is the interval evaluation of $x - x$ when x lie in the range $[-1, 2]$: $[-1, 2] - [-1, 2] = [-1, 2] + [-2, 1] = [-3, 3]$. However this overestimation will decrease with the width of the ranges and there are several methods that may reduce the effect of the wrapping: some of them will be described later on. Note also that if an expression involves only a single occurrence of each unknown, then the interval evaluation is exact in the sense that A, B are respectively the real minimum and maximum of F (however up to the round-off errors).

Using this point it is simple to design an interval analysis solver that will be described for the forward kinematic problem. First we have to introduce the concept of *box* which is basically a Cartesian box the the 6-dimensional parameter space $\{k_1^r, \phi_1^r, \psi_1^r, k_2^r, \phi_2^r, \psi_2^r\}$ with a range for each of the variable. In our case we are looking for a solution in the box $\mathcal{B}_1 \{[-1, 1], [-\pi/2, \pi/2], [0, 2\pi], [-1, 1], [-\pi/2, \pi/2], [0, 2\pi]\}$. Using the interval evaluation we may design a procedure $\mathcal{T}(\mathcal{B}_i)$ that returns 0 if a given box \mathcal{B}_i does not satisfy one of the constraint. A box \mathcal{B}_i may also be *bisected*: we select one of the box range (e.g one having the largest width) and

split the box along this direction at the mid-point of the range, which will create 2 new boxes that will be added at the end of \mathcal{L} . Our algorithm will maintain a list \mathcal{L} of box that is initialized with \mathcal{B}_1 . During the process we will add boxes to this list: n will denote the total number of boxes in the list while j will be the current box number.

```

initialization,  $\mathcal{L} = \{\mathcal{B}_1\}, j = 1, n = 1;$ 
while  $j \leq n$  do
    if  $\mathcal{T}(\mathcal{B}_j) = 0$  then
        |  $j = j + 1$ , next;
    end
    if  $\text{width}(\mathcal{B}_j) \leq \epsilon$  then
        |  $S = S + \mathcal{B}_j$ ;
        |  $j = j + 1$ , next;
    end
    Bisect( $\mathcal{B}_j$ );
     $n = n + 2, j = j + 1$ ;
end
    
```

In this very basic algorithm a box of width lower than a threshold ϵ is considered to include a solution. Such an algorithm is however usually inefficient. A clear improvement is to manage the list differently to avoid having a very large list. Let B_1, B_2 be the 2 boxes resulting from the bisection: B_1 will be stored as \mathcal{B}_j while we shift by one the boxes $\mathcal{B}_{j+1}, \dots, \mathcal{B}_n$ by 1 and store B_2 as \mathcal{B}_{j+1} so that the bisection add only a single box to the list while we are fist managing smaller and small boxes until either the current box is eliminated or stored as a solution. This algorithm is guaranteed to complete (because of the test on the box width) and cannot miss a solution but may induce spurious solution. For avoiding this problem we use the Kantorovith theorem [?]: without going into the details this theorem allows to prove that, provided that some constraints on the norm of the Jacobian and Hessian matrices are satisfied (the *Kantorovitch constraint*), then there is a single solution in a given box and ensure that the Newton scheme will converge toward this solution. This addition allows us to determine exactly (in the sense that we are able to determine an approximate solution whose distance to the exact solution is arbitrary low) all solutions. At the same time a box that satisfy the Kantorovitch constraint will just be discarded from the list as soon as the solution has been computed with the Newton scheme. Consequently the Kantorovith theorem will improve also the efficiency of the algorithm but we may still add some improvements. First in general the more complex an equation is the larger will be the overestimation of its interval evaluation and it is surprisingly usually efficient to improve the efficiency of an IA algorithm by using more unknowns but simpler expressions. For example in our case adding the R^r as additional unknown and equation (8) as additional constraint allows to reduce the computation time of the basic algorithm by a factor of 100.

Another improvement is based on the rewriting of the equations. Consider for example equation (6) that be written first as $k^r \sin \phi^r = \cos \frac{\psi^r}{2}$ from which we deduce (provided that the interval evaluation of $\sin \phi^r$ does not include 0) $k^r = \cos \psi^r / 2 / \sin \phi^r$. The interval evaluation of the right hand term is then intersected with the range of k^r and we assign to k^r the intersection range. Another example may be provided for the inverse kinematic. First we may assume that the interval evaluations of the x_r, y_r as provided by equations (4, 5) are largely overestimated. In the current process we multiply these ranges by \mathbf{T}_r^F which is an interval matrix and thereby increases the overestimation for x_p, y_p . A more efficient manner to deal with theses equations is to compare directly x_r, y_r with the result of $(\mathbf{T}_r^F)^{-1} \bar{\mathbf{p}}_p = (x_a, y_a)^T$. We may then notice that $\sin \psi^r x_r - \cos \psi^r y_r = -2U k^r (\cos \phi_2 - \cos \phi^r)(2) = \sin \psi^r x_a - \cos \psi^r y_a(3)$, where $U = \sqrt{EI/R}$. The constraint (2)=(3) involves only simple expressions with no overestimation as there are on single occurrences of the unknowns. They may also be used for rewriting, for example by isolating the value of k^r, R for a possible update.

Another efficient method to improve the interval evaluation is to consider the derivatives of the expression. Consider for example the constraint (8) that is used to update the value of R . As k^r has two occurrences in this expression, then there maybe an overestimation of R . We consider the expression $\mathcal{F} = F(k^r, \phi_2) - F(k^r, \phi^r)$ and its derivative with respect to k^r of which we compute an interval evaluation $[a, \bar{a}]$. If $a > 0$, then \mathcal{F} is increasing with k^r and the interval evaluation of \mathcal{F} is established as $[Min(\mathcal{F}(\underline{k}^r, \phi^r), Max(\mathcal{F}(\bar{k}^r, \phi^r))]$: in these expression ϕ^r appears only once and therefore we get an exact interval evaluation of \mathcal{F} and therefore of R . A similar result will be obtained if $\bar{a} < 0$.

Another improvement of the algorithm is to run a few iteration of the Newton scheme for each processed box, using the mid-point of the box as initial guess. This may allow to quickly find solution of the kinematic problem and to discard boxes that include the discovered solutions as soon as they satisfy the Kantorovitch conditions.

Nevertheless there is a computing price to pay for the guarantee of finding all solutions. We have implemented solving algorithms based on IA in Maple for both the inverse and forward kinematics. The computation time of the forward kinematic is between 30 minutes and one hour, while it takes about 12 hours to solve the inverse kinematic. Being given that Maple is much slower than a C++ version of interval arithmetic we estimate that the computation time of a C++ version of our algorithm will be around 4 minutes for the forward kinematic and 1 hours for the inverse kinematic. The IA version is therefore much slower than the solving method proposed in the previous section. Note however that a distributed version of the IA algorithm is easy to implement as the processing of a given box is independent from the processing of another box. A distributed implementation induces however an overhead because of the transmission of the box elements between the computers but as the size of the data that have to be transmitted is small this overhead is relatively small. Previous test for kinematic

problems have shown that as rule of thumb we may assume that the computation time will be divided by $n - 2$ when using $n \geq 10$ computers.

There is however a possible strategy for again improving the efficiency of the IA algorithm. We will first introduce the new variables

$$\begin{aligned} U_1 &= E(k^r, \phi_2) & U_2 &= E(k^r, \phi^r) \\ V_1 &= F(k^r, \phi_2) & V_2 &= F(k^r, \phi^r) \end{aligned}$$

that will also be added as constraints. With these new variables equations we get

$$R^r = \frac{EI}{L_r^2} (V_2 - V_1)^2$$

while equations (4, 5) may be written as

$$\begin{aligned} x_r &= -V(\cos \psi^r (2(U_1 - U_2) - 2(V_1 - V_2) - 2k^r \sin \psi^r (\cos \phi_2 - \cos \phi^r)) \\ y_r &= -V(\sin \psi^r (2(U_1 - U_2) - 2(V_1 - V_2) - 2k^r \cos \psi^r (\cos \phi_2 - \cos \phi^r)) \end{aligned}$$

where $V = \sqrt{EI/R}$. As the unknowns have all a single occurrence there will be no overestimation of the interval evaluation of x_r, y_r . Furthermore using this form it is easy to obtain one of the unknown (say U_1) of the U_1, U_2, V_1, V_2 as a function of the other U_i, V_i and of x_r, y_r . As seen previously this rewriting may allow to update the range of U_1 .

But the larger advantage is the management of the buckling mode. Up to now the algorithms that have been described in the previous sections were assuming a given buckling mode for each of the rod with the consequence that if we are looking for the kinematic solutions for the rods that may have any buckling mode say among 1, 2 and 3, then we have to run our algorithm $3^9 = 9$ time. Assume now that in our IA boxes we add a new discrete variable that is the buckling mode, initialized with the set $[1, 2, 3]$. The buckling mode plays a role only in U_1, V_1 and V but as $E(k^r, 5\pi/2) = E(k^2, \pi/2)$, $E(k^r, 3\pi/2) = -E(k^2, \pi/2)$ and $F(k^r, 5\pi/2) = F(k^r, \pi/2)$, $E(k^r, 3\pi/2) = -F(k^2, \pi/2)$ the interval evaluation of x_r, y_r taking into account the possible multiple buckling modes require only a single evaluation of U_1, V_1, V . The update of these variables in the IA algorithm will progressively allow us to discard one buckling mode until a single one remain. With this modification we will get a single algorithm that will allow to find all solutions for any pair of buckling mode while avoiding a large overhead due to repeated interval evaluations of the same $E, F(k^r, \phi^r)$ if we were using a sequential approach.

6 Discussion

Clearly our solving methods cannot be used in a real-time process where we have a good initial guess of the solution. But this is not a problem as in this case we may use a certified Newton scheme based on a mix of Kantorovitch theorem, IA and Newton as described in [?].

(JPM comment: Here you may possibly explain why your solving will not allow to manage 3 rods, either in the plane or in space ?)

If we assume now 3 rods in the plane connected not at the same point we will have to solve a square system with 12 unknowns (the set of k, ϕ, ψ for the rods and the end-effector pose parameters) for the forward kinematic and 12 unknowns (the set of α, k, ϕ, ψ for each rod) for the inverse kinematics. Although we have not tested the solving with IA, we believe that these problems are tractable.

(JPM comment: Perhaps you can generalize for spatial robots with n rods ?)

An advantage of the IA algorithm for the forward kinematic is that we may take into account bounded measurement errors on the α . In that case the IA algorithm will provide regions constituted of a list of Cartesian boxes that are guaranteed to include the real pose of the robot.

(JPM comment: shall we mention the test done on the continuation method here or keep them for another paper ? I believe we should but this is up to you)

7 Conclusions

(JPM comment: A comment on the number of solutions ?)

A classical possibility to simplify kinematics problems is to add information for solving the forward kinematic. For example we may add a sensor that will measure the angular distance $\psi_2 - \psi_1$ between the tips of the rod at the end-effector. We may then add to the IA algorithm the constraint $\psi_2 - \psi_1 \in [m - \epsilon, m + \epsilon]$ where m represents the measurement and ϵ the measurement error. Clearly this additional constraint will considerably speed-up the solving (because of the elimination of the boxes that will not satisfy this constraint) but it is unclear if this will always lead to a single solution. Furthermore if we assume also measurement errors on the α this additional information may sharpen the possible regions for the end-effector pose.

(JPM comment: Another possibility will be to have a bending sensor say at the mid-length of the rods but I am unsure about the corresponding constraint equation. We may also add passive cable length sensor that will provide a good estimation of the pose.)

Acknowledgments

The authors received financial support from the Spanish Government (DPI2015-64450-R) and the Regional Government of the Basque Country (Project IT949-16).

References